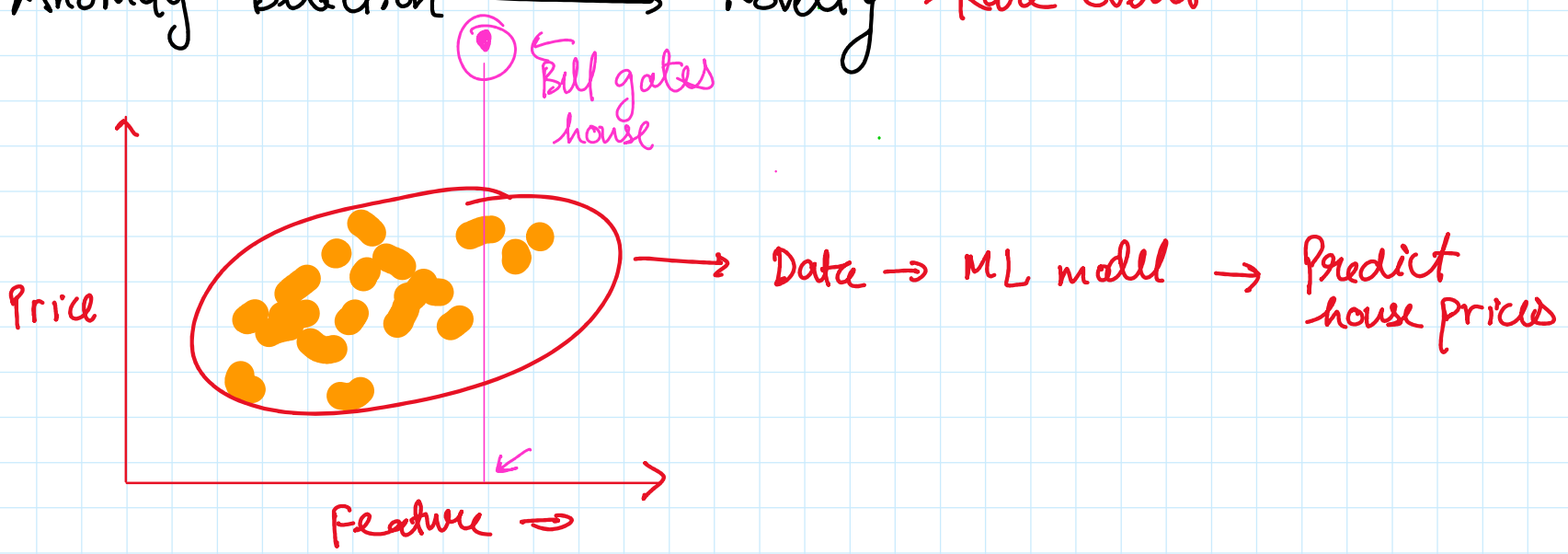
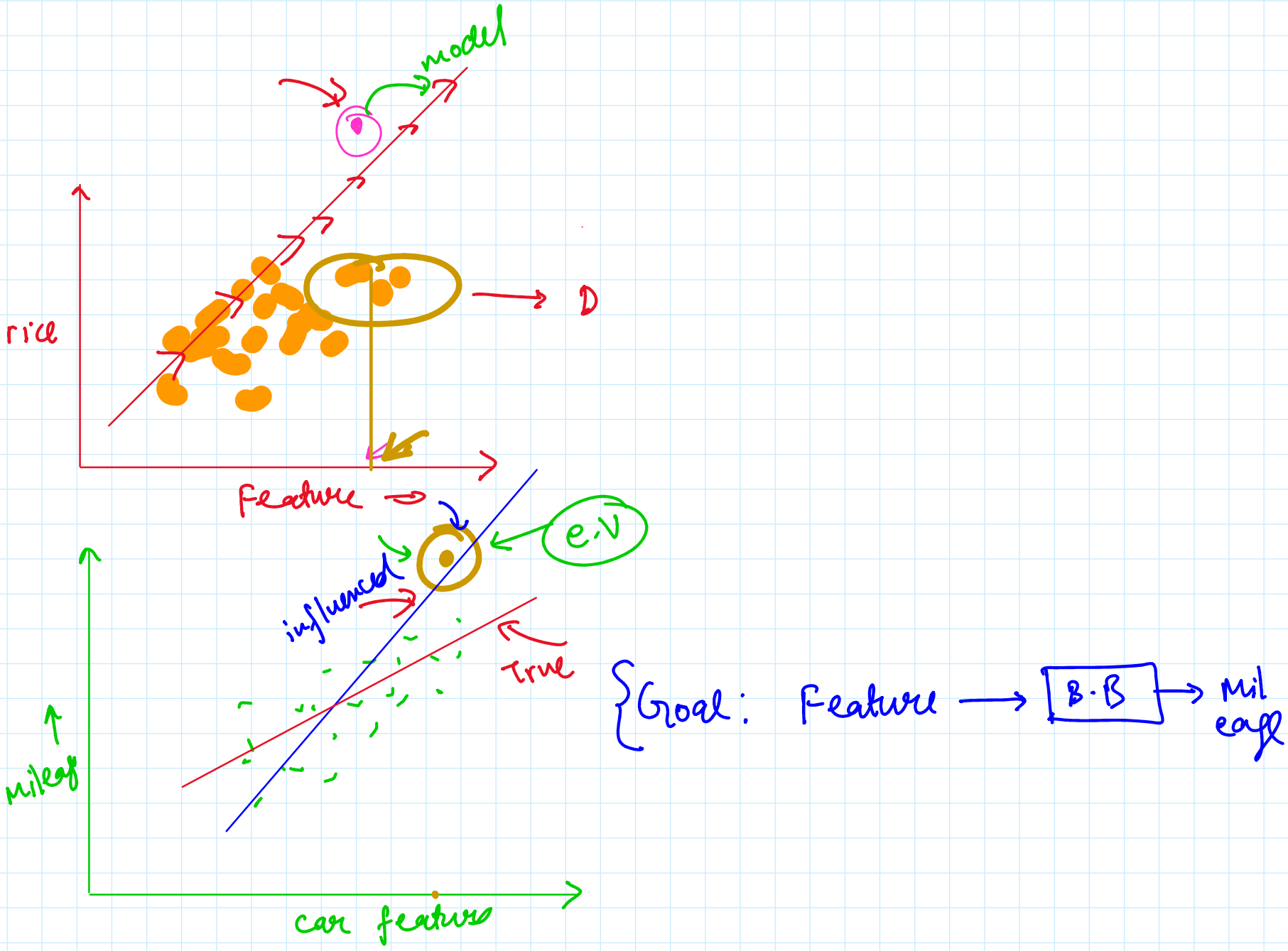
Anomoly Detection

19 February 2024 20:47

Anomdy Detection → Outliers → "wrong" data

Novelty → Rare event





{Goal: Feature → B·B → Mileage

How ? → Algorithm → Set of Rules → Cleaned Data

Agenda → {Elliptical Envelope}

→ Isolation Forest

→ LOF (Local outlier Factor) (Drive)

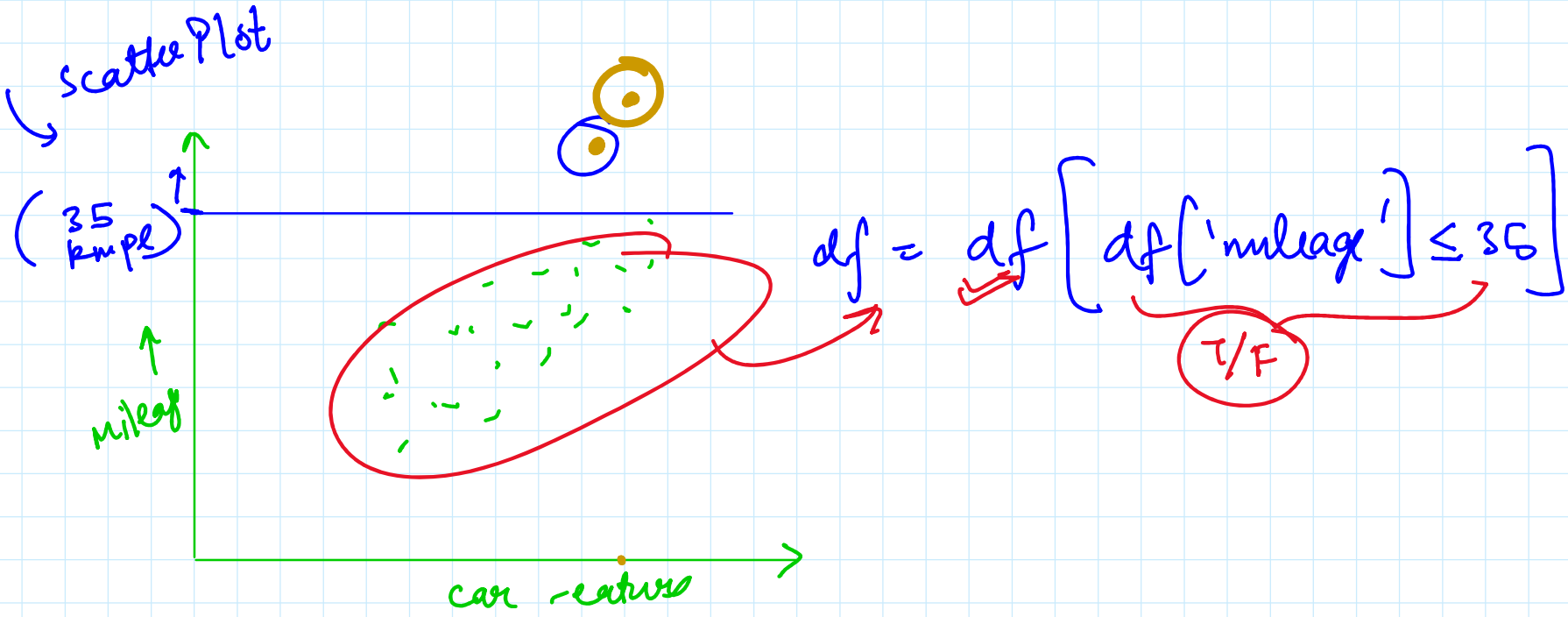
Simple Methods

→ Box Plot

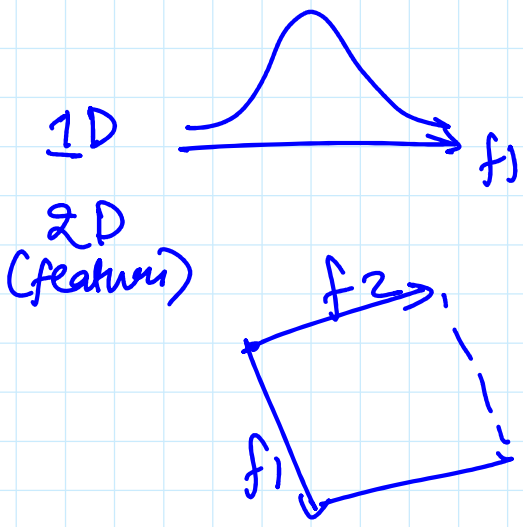
→ Z score → $Z = \frac{x-\mu}{\sigma}$ → 3.2 [Z = 3]

→ Visualise (Remove manually)

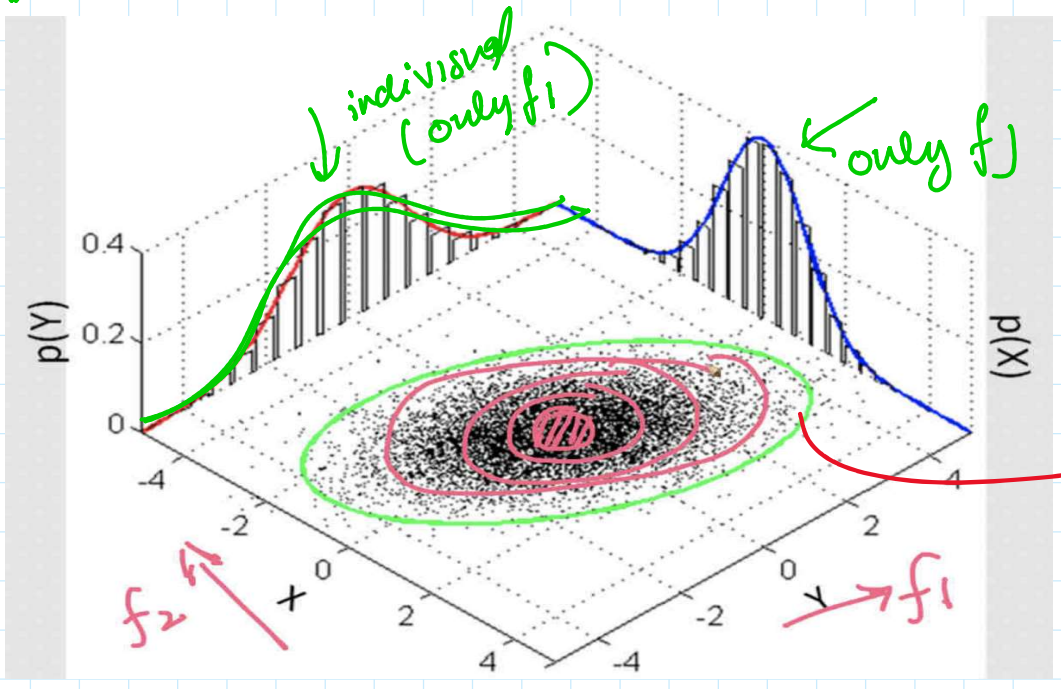
↳ (Scatter Plot → Filter Data)



Elliptical Envelope



*(Combined)



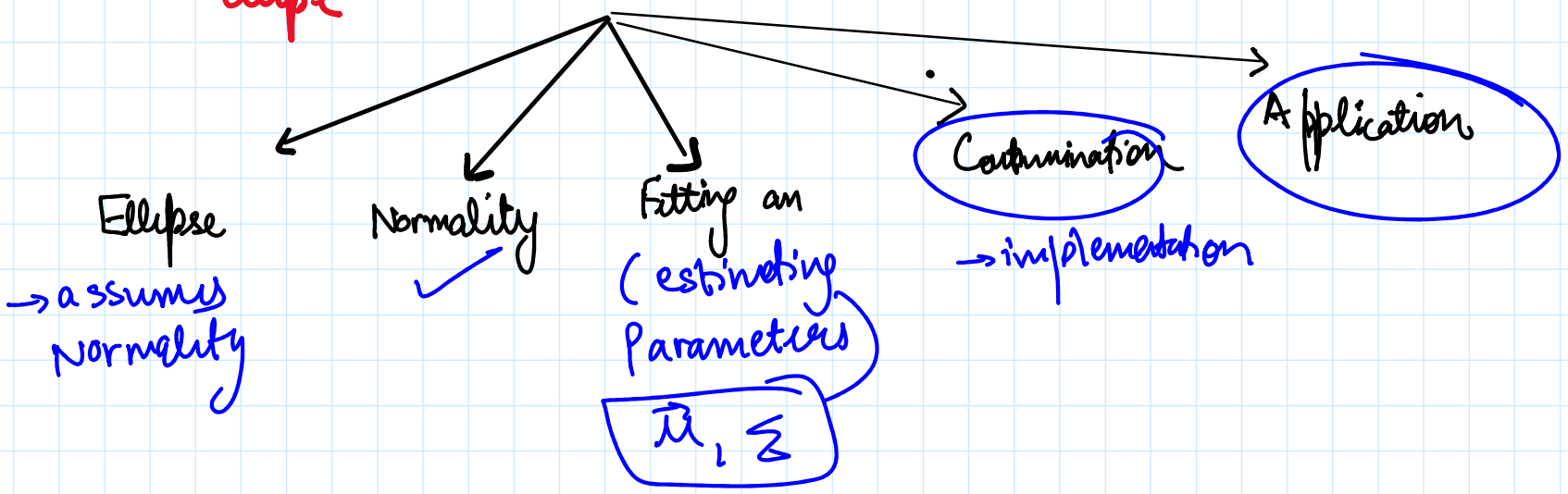
(Joint Distrb^n)

assume (Data → Gaussian)

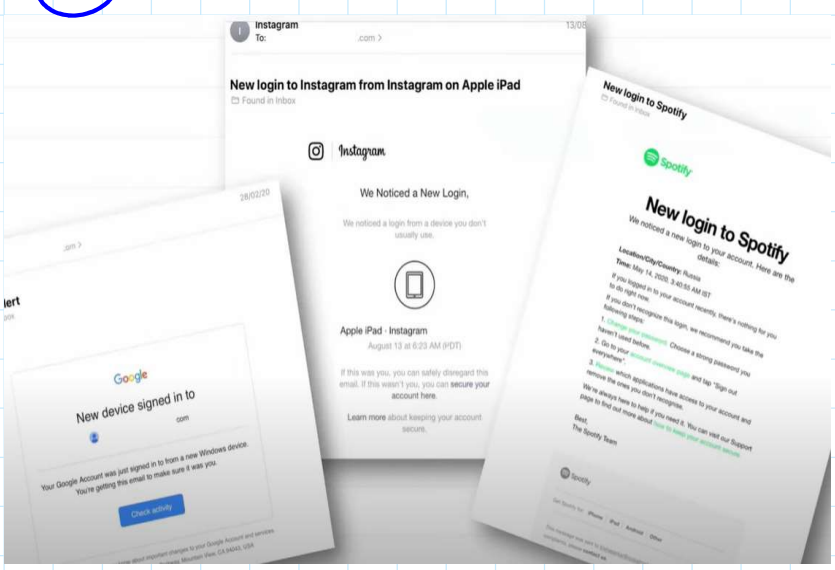
Elliptical Envelope → statistical / multivariate / outlier Detection (for) / Gaussian assumes

↳ fits ellipse to data and points are removed which are not in that ellipse

ellipse



(X) Blocks that request



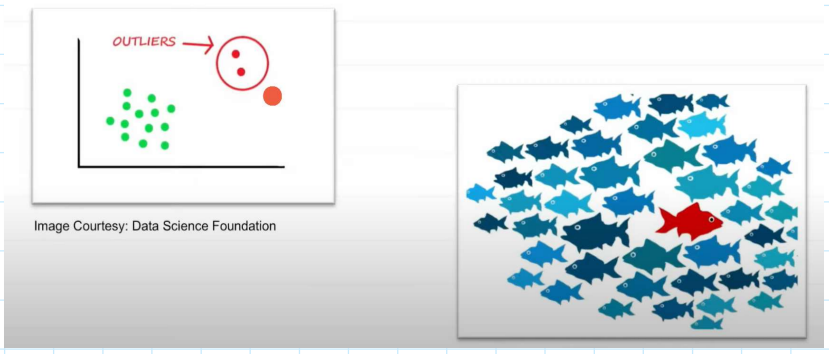
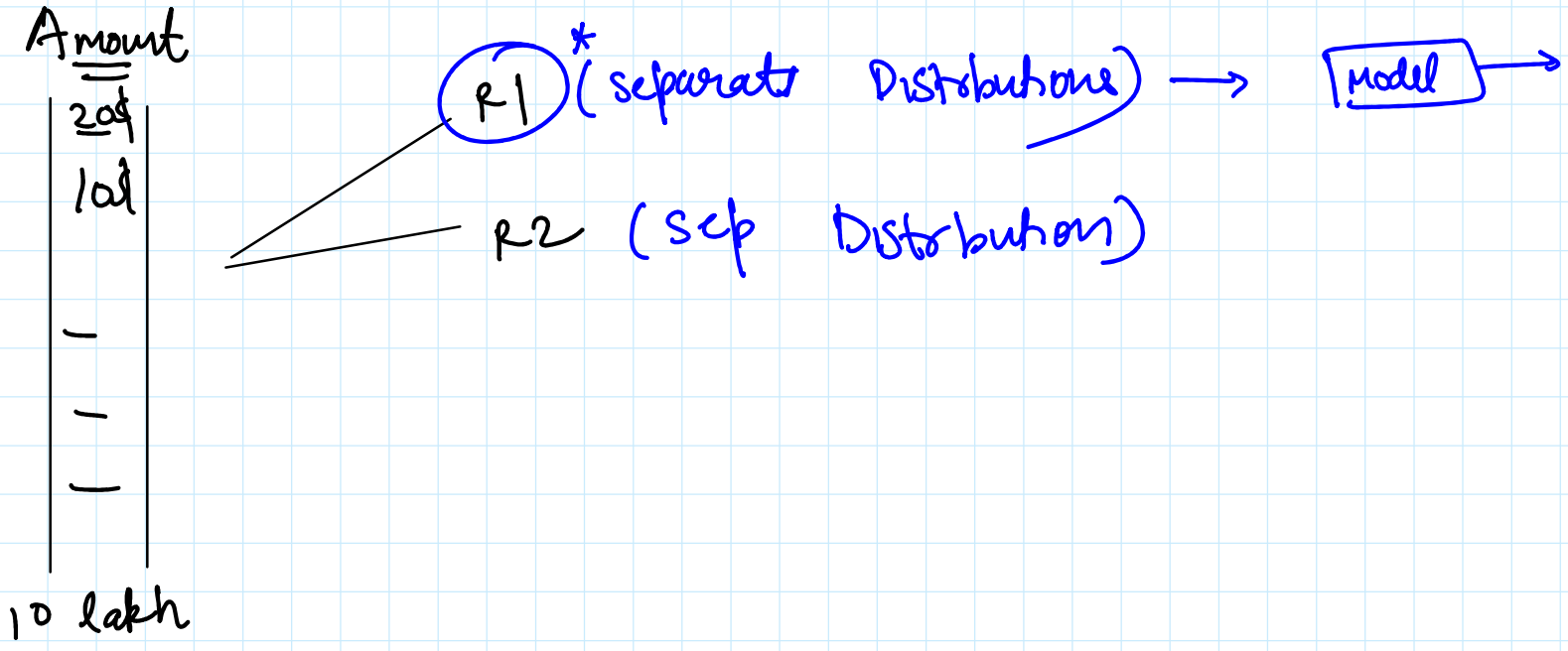


Image Courtesy: Data Science Foundation

Threshold = $Z_\alpha \rightarrow 3$ $(z \geq 3)$ (outlier)

Payments → (Red money) $Z \rightarrow$ < 3 (safe)
> 3
$= 3$
(you dont know)

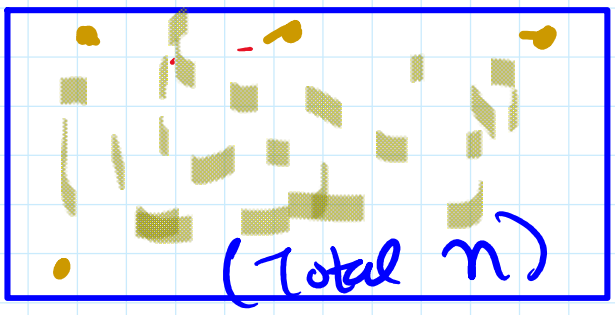


Fit a Distribution (one Distribution) → Dimension is high
→ (Data) (we have) → fit distrb^n

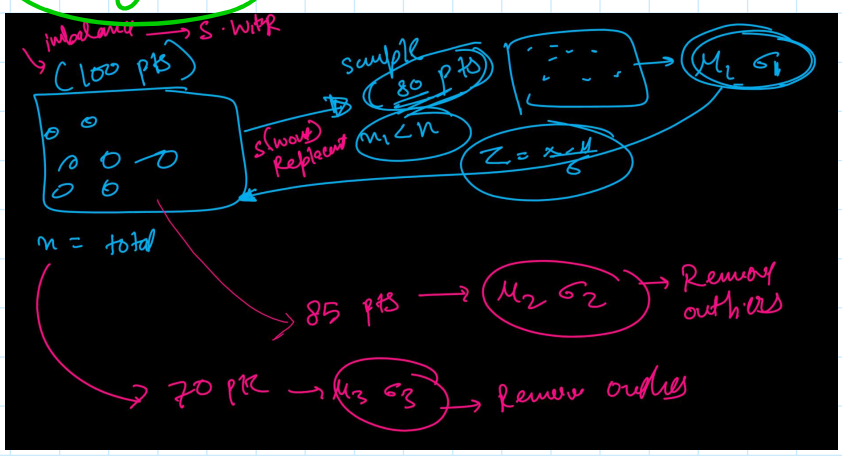
New Point → Z score → threshold → Anomaly

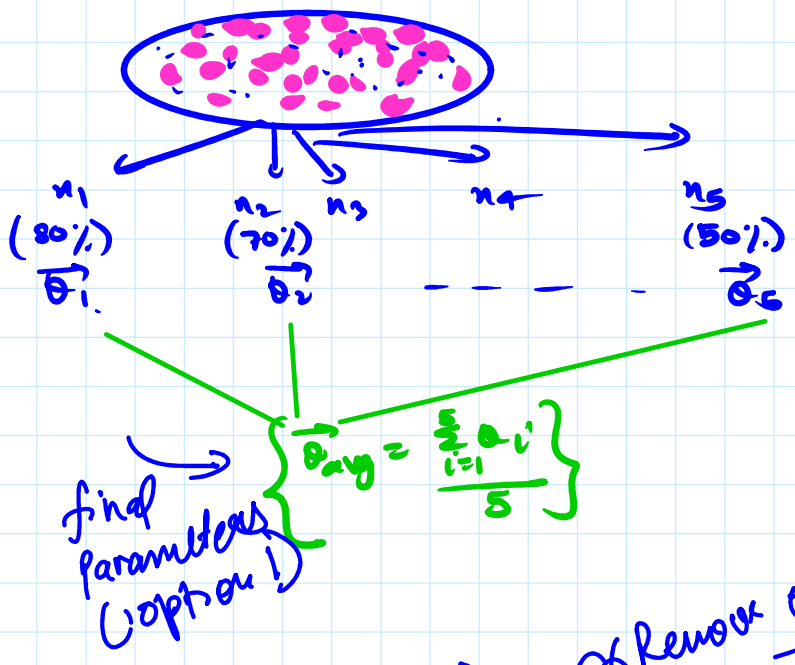
* RANSAC

RANSAC: RANSAC is an iterative method used primarily in computer vision and machine learning to estimate parameters of a mathematical model from a set of observed data that contains outliers. It's particularly useful when you expect a significant fraction of the data to be outliers.

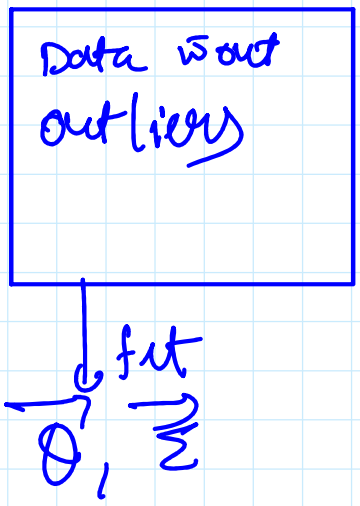


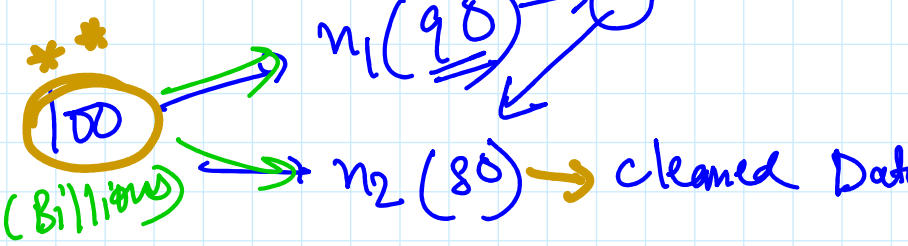
Rough

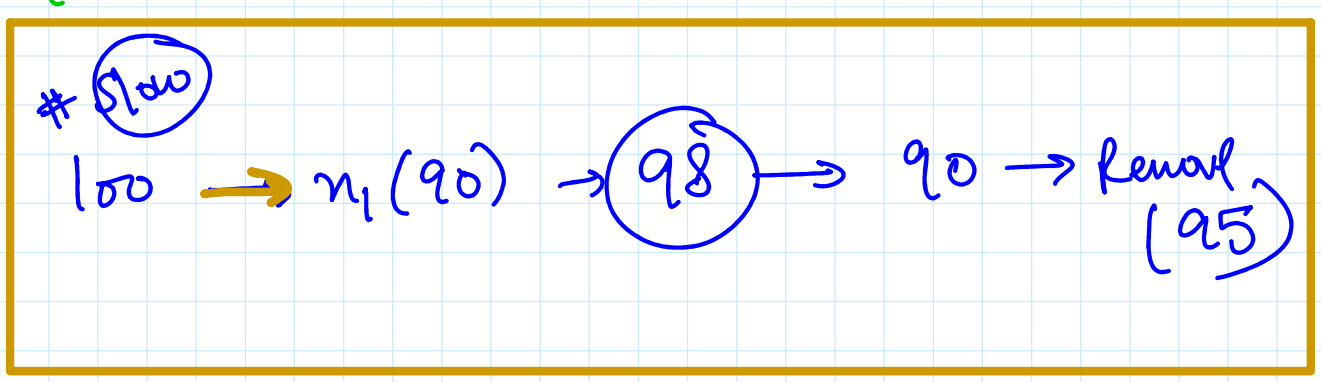




option (II)







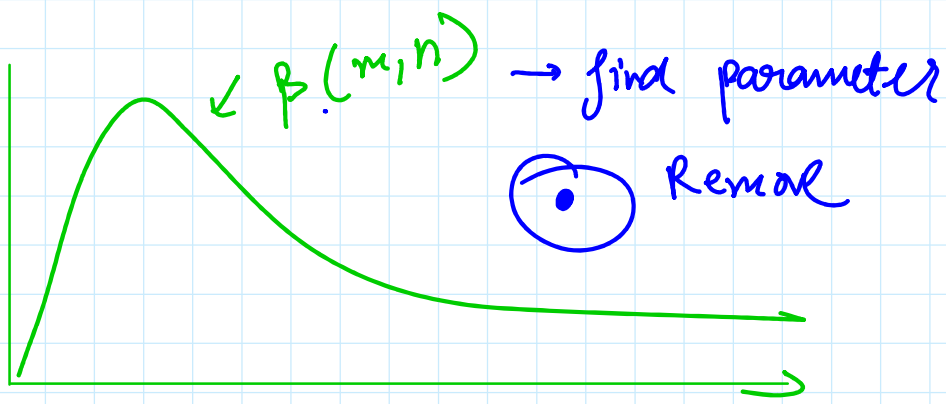
E.E assume Gaussian → convert $X \rightarrow \log(x) = Y$

$Y \sim N(\mu, \sigma)$

Q. For non-gaussian, do we need to convert into gaussian distribution?

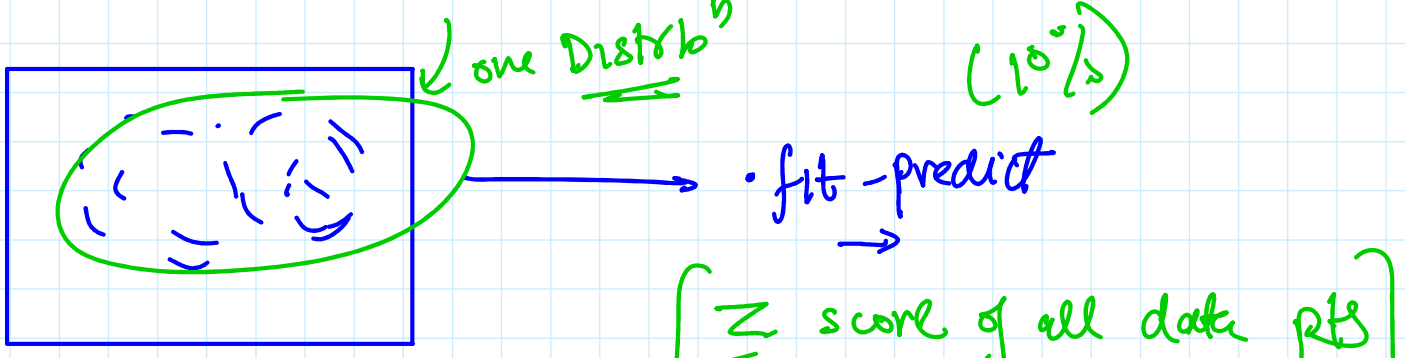
- While the elliptical envelope method makes an assumption that the distribution is gaussian, the strategy can be applied to any distribution
- As long as we know any distribution and its parameters θ, we can extend our strategy to use RANSAC and estimating parameters θ
- The can be other distributions such as multivariate poissons, multivariate log normals, etc., but we dont use them as much
- One other strategy is to convert them into gaussians but we dont necessarily have to.

- The can be other distributions such as multivariate poissons, multivariate log normals, etc., but we dont use them as much
- One other strategy is to convert them into gaussians but we dont necessarily have to.
- As long as there is any distribution we can calculate the probability of $x_i \in X$ using PMF/PDF



⊕ Sk-Learn → (Package) • fit • predict
labels the outliers

- Contamination → 0.1 → (10%)
- assumed_centered → True → Standardised / centered
- Elliptical → (multivariate Gaussian)



Disadvantages →

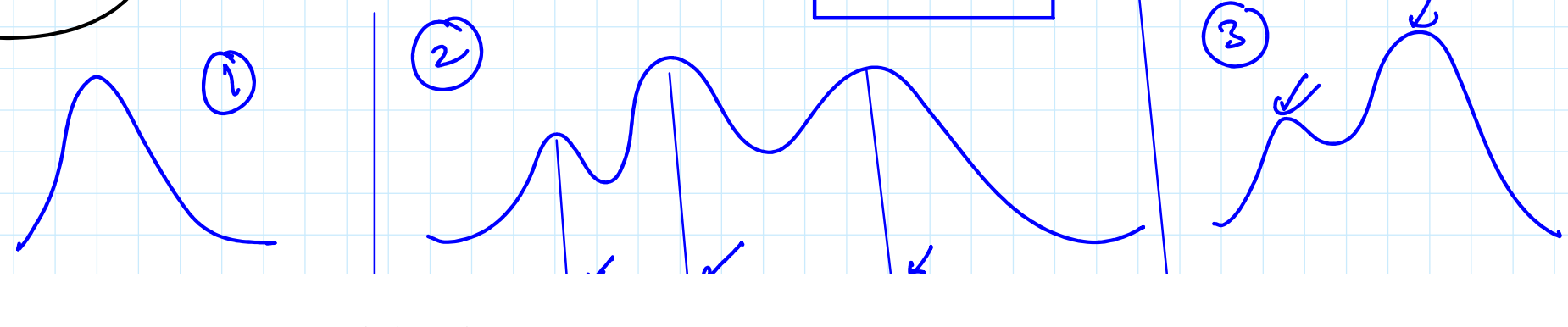
Disadvantages

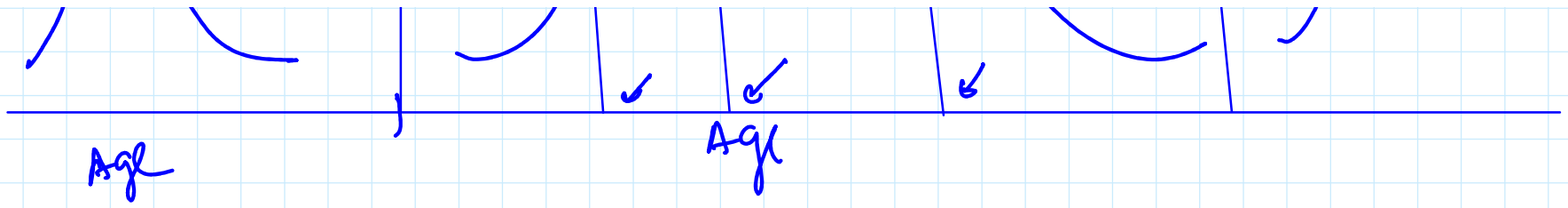
You might have find the concept of elliptic envelope quite straightforward, but there are limitations.

- **It cannot be used non-unimodal data**
- It is specifically for **multivariate gaussians**

If the data fails to meet the assumptions of unimodal and multivariate gaussian, the whole things crashes

Break → 7 mins → start at 22:20

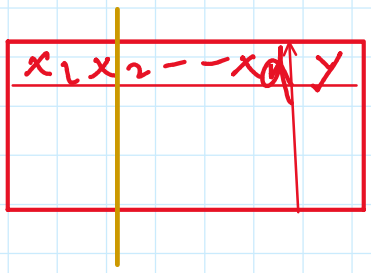




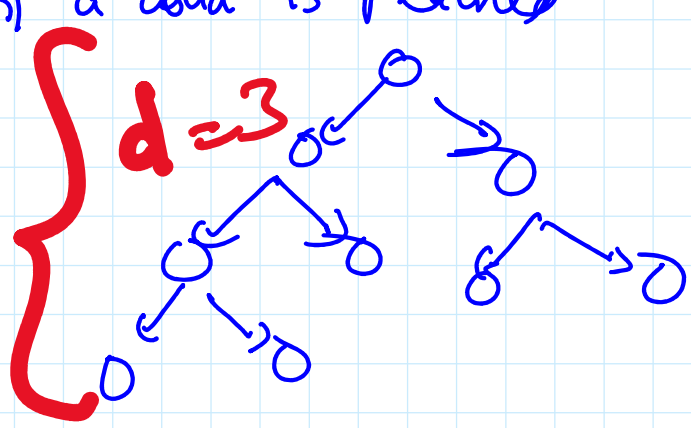
Isolation Forest (Categorical + Numerical)

(#) Concept → Behaviour → outliers, Inliers } is Different

D-Trees → supervised → X → Y

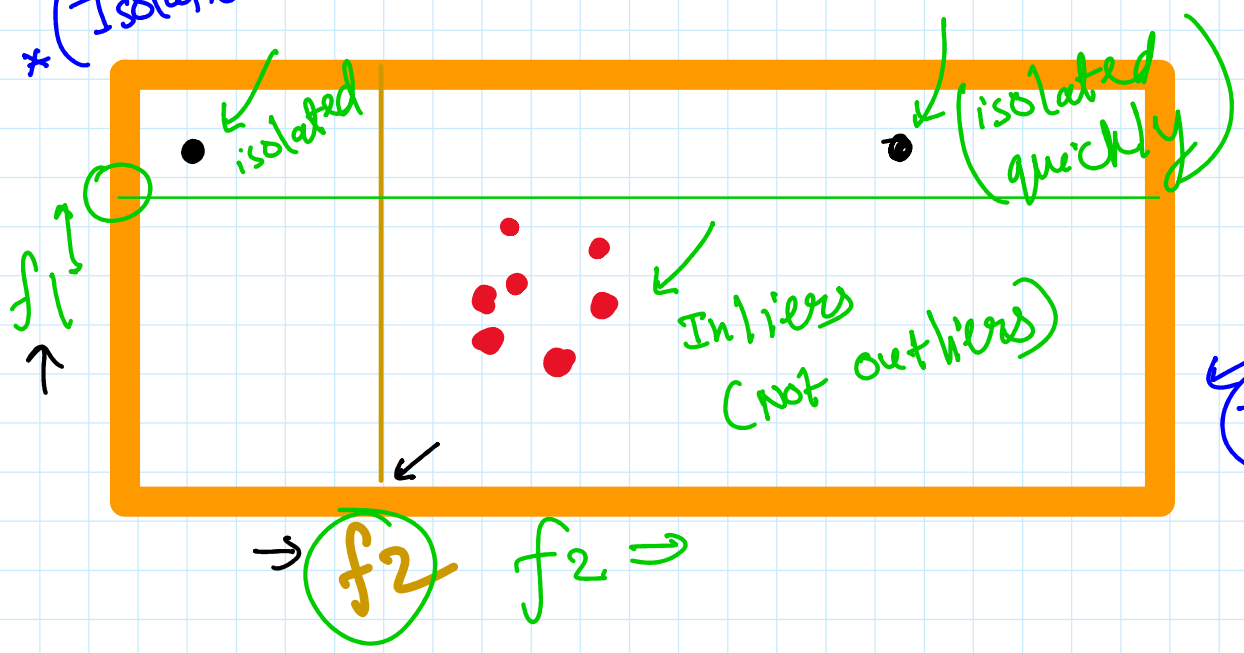


→ Choose "best" feature
→ Choose & split threshold
→ keep splitting repeatedly
→ until a cond" is reached



Tree → Forest ?

*(Isolation Forest)



Logic → outliers are isolated easily than inliers

Unsupervised → No Y (Target)

Isolation Forest Algo

S1 → Randomly ! pick a feature

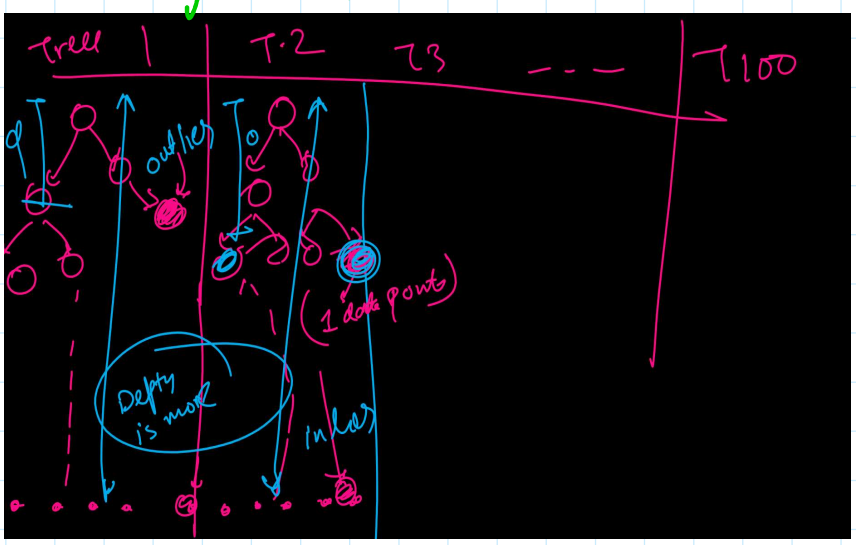
S2 → Randomly ! threshold that feature

S3 → Split (Build) each tree until the leaf consist of only one data point (condition)

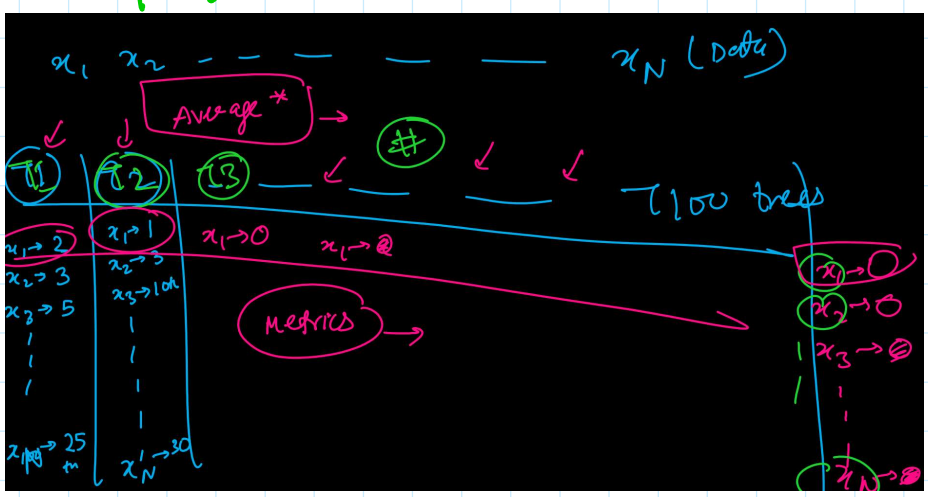
S4 → Build many Trees (forest)

(S4) → Build many Trees (forest)

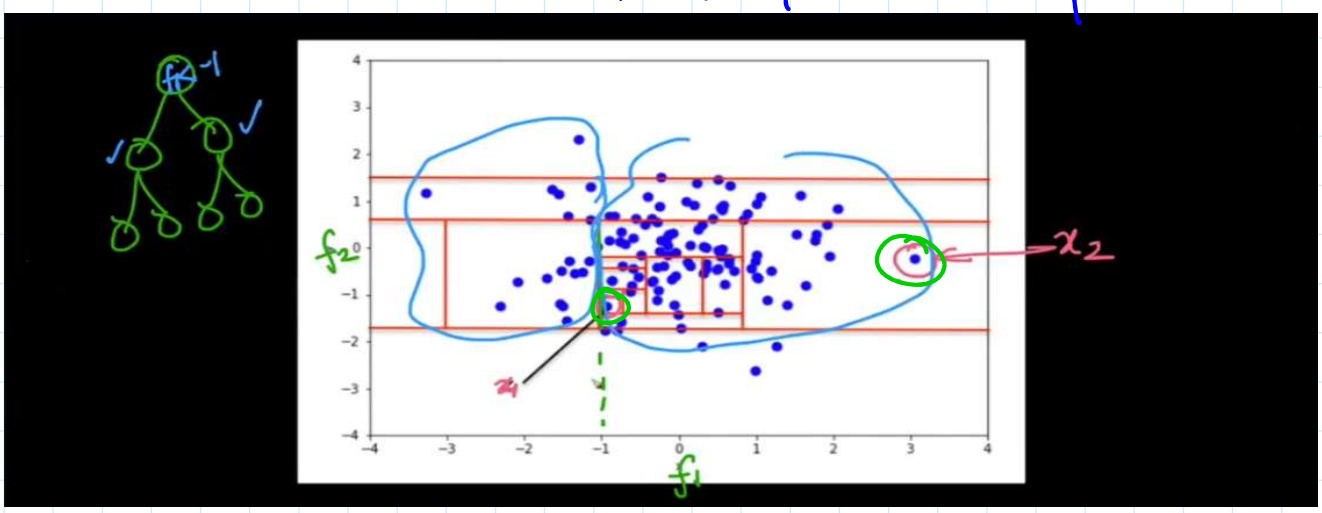
Rough (2)



Rough (3)



Axis Parallel split Problems



. What is the whole idea behing Isolation Forest?
So, to sum it up, the idea behind Isolation forest is,

- **On an average outliers have lower depth in the random trees**
- **On an average, inliers have more depth in the random trees**

How can we evaluate Isolation Forests?

- Imagine, we have build 100 random trees. For each point □□*xi* in the dataset, we can get an average depth.
- We use this average depth to convert into a metric.
- Apart from this, there are lot of different metrics, that people have came up with over the years
- But, the basic intuition is that lesser the average depth, higher likelihood is there that it is an outlier

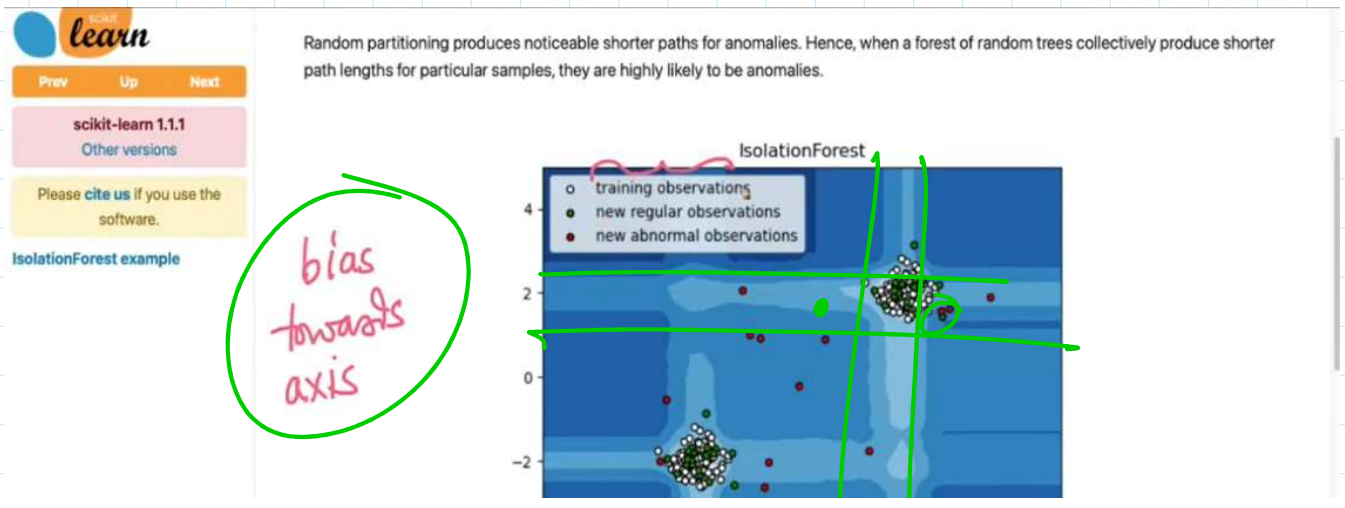
But, how do we decide average depth for a point to be classified as an outlier?

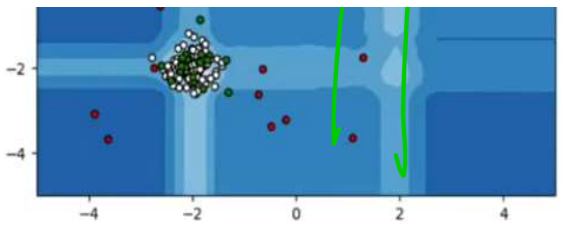
- There is no one metric specifically used for average depths in iForests. At the end, whichever metric you use, it is based on the threshold.
- There are a lot of metrics that researchers have came up with over the years.
- But, studying them in this lecture is out of scope.

* (sample some rows)

But, what if the number of datapoints is large? Wouldn't it mess up the Isolation Forests?

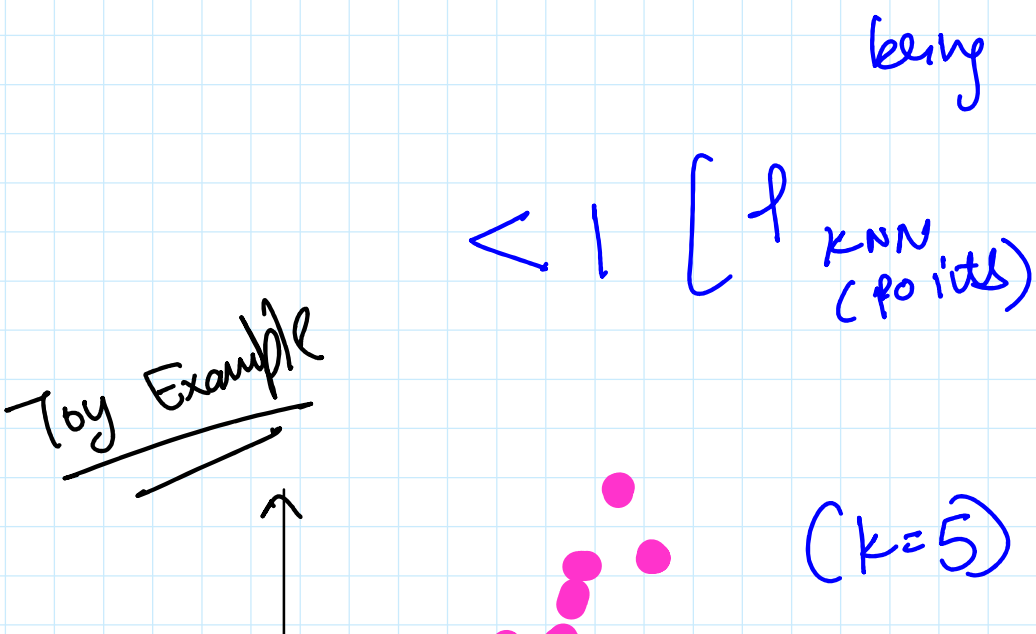
- iForests can be made on subset of samples.
- We use this subset as train dataset and the rest of the data as test datase





D.T → Axis Parallel Splits → Cannot capture non-linear complex relationships → Axis Bias

LOF → Local Outlier Factors



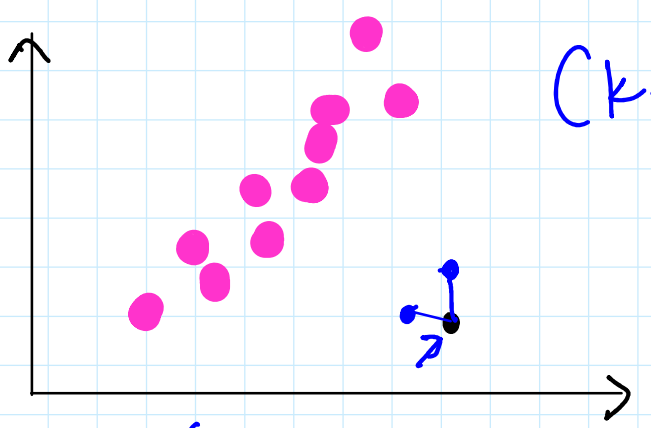
LOF = Lof = (Ratio of density of KNN) / to density of observation itself

Lof > 1 [ρ KNN(Points) > ρ point*] ⇒ Neighbours are dense

⇒ (Point has possibility of being outlier)

< 1 [ρ KNN(points) < ρ point*] ⇒ that point is not outlier

Toy Example



(k=5)

Lof = Ratio = NR / DR

Density ● = area of ○ with 5 nearest observation

(*) ∴ Density ● = area of ○ with radius = their 5th nearest neighbour

(*) inverse { Density ↑ ● = area of ○ with radius = their 5th nearest neighbour
↓ ● =
● =
● =
● =